

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

BAKALÁŘSKÁ PRÁCE

2008

Vladimír Spál

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Návrh a zavedení databázové aplikace pro evidenci obchodních případů

The Design and Installation of the Database Application for Record of Business Transactions

Student: Vladimír Spál

Vedoucí bakalářské práce: Ing. Vítězslav Novák, PhD.

Ostrava 2008

„Místopřísežně prohlašuji, že jsem celou práci včetně všech příloh vypracoval samostatně.“

.....

Pod'akovanie:

Ďakujem pánovi Ing. Vítězslavovi Novákovi, PhD. za odbornú pomoc pri vedení mojej bakalárskej práce. Taktiež ďakujem pánovi Ing. Břetislavovi Duchovi za poskytnutie informácií, ochotu a čas, ktorý mi venoval pri spracovaní mojej bakalárskej práce.

Obsah:

Úvod	1
1 Teoretické východiská tvorby databázových aplikácií.....	3
1.1 Relačná databáza	3
1.1.1 Normalizácia.....	5
1.1.2 Integritné obmedzenie	6
1.2 Microsoft Access	7
1.2.1 Tabuľky	7
1.2.2 Dotazy	8
1.2.3 Formuláre	9
1.2.4 Zostavy	9
1.2.5 Ostatné objekty	10
1.3 Vytvorenie aplikácie v Microsoft Access.....	10
1.3.1 Vytvorenie aplikácie bez pomoci sprievodcu	11
1.4 Visual Basic for Application	13
1.4.1 Výhody programového kódu VBA.....	14
2 Analýza súčasného stavu	15
3 Návrh a zavedenie databázovej aplikácie.....	17
3.1 Dátová štruktúra	17
3.1.1 Popis a štruktúra tabuliek	17
3.1.2 Relácie medzi tabuľkami	21
3.2 Návrh formulárov	22
3.3 Návrh zostáv	27
3.4 Dokončenie aplikácie	28
3.5 Zavedenie aplikácie u zákazníka	28
4 Zhodnotenie prínosov navrhutej databázovej aplikácie.....	29
Záver	31
Zoznam použitej literatúry:.....	33
Zoznam skratiek a symbolov	34

Úvod

Každý podnik dospeje do stavu, keď veľkosť obratu, zložitosť procesov a túžba po hľadaní zdrojov ďalšieho rastu postaví manažérov pred otázku, či zaobstarat' riešenie Business Intelligence.

Veľké množstvo organizácií pociťuje potrebu vidieť informácie získané ako interne v spoločnosti, tak aj iným pohľadom. Je tu trend zviditeľňovania informácií, ktoré môžu viesť k rýchlemu rozhodnutiu. Toto rozhodnutie môže viesť k poskytovaniu viacerých služieb alebo informácií. Významnú rolu tu zohráva faktor kvality a výkonnosti podnikového riadenia, teda konkurencieschopnosť spoločnosti. [7]

Systémy, ktoré sú nazývané Business Intelligence Systems, transformujú dáta a znalosti do dát štruktúrovaných. Pojem Business Intelligence nie je potom nič iného ako sada konceptov a metód určených pre skvalitnenie rozhodnutí spoločnosti. Inak povedané BI predstavuje komplex aplikácií, ktoré podporujú analytické a plánovacie činnosti podnikov a organizácií. [7]

Cieľom je navrhnuť a implementovať manažérske aplikácie, dátové sklady a trhoviská, ktoré spracúvajú analytické a rozhodovacie úlohy v riadení spoločnosti so zodpovedajúcou organizačnou a technologickou infraštruktúrou. Tieto systémy pomáhajú riešiť problémy informačných, transakčných systémov a teda vytvárajú priestor pre skvalitnenie riadenia spoločnosti. [7]

Jednou zo spoločností pôsobiacich na českom poisťovacom trhu je BD FINANCE a.s. so sídlom v Ostrave. Ich hlavnou pracovnou náplňou je poskytovanie služieb poisťovacieho makléra. Jedná sa o nedávno založenú akciovú spoločnosť, v ktorej momentálne pôsobí jeden poisťovací maklér.

Aby som priblížil pojem poisťovací maklér: Každý rozumný človek dnes nemárni čas a energiu na činnosti, ktoré je lepšie prenechať odborníkom v danej oblasti. Právne služby je lepšie nechať na bedrách právnikom, obchody na burze burzovým maklérom a poistenie poisťovacím maklérom. Aj na českom poisťovacom trhu totiž medzi stranou predávajúcich (poisťovne) a kupujúcich (občania a firemná klientela) existuje medzičlánok, ktorý obchod medzi nimi dokáže sprostredkovať. Tými sú už zmienení poisťovací makléri. Ich činnosť je totiž založená na podobnej platforme ako u finančných poradcov. Poisťovací maklér pracuje na základe zmluvy, poprípade plnej moci uzavretej

s konkrétnym klientom, ktorého voči poisťovniam zastupuje. Teda nepredáva produkty jednej konkrétnej poisťovne, ale vyhľadáva pre záujemcov o poistenie najvýhodnejšiu ponuku poisťovní na trhu. [8]

Ako z náplne práce spoločnosti BD FINANCE a.s. vyplýva, aj v tejto spoločnosti vznikla potreba používania nejakého nástroja Business Intelligence. Potreba zahŕňa najmä evidenciu prejednávanych obchodných prípadov, existujúcich obchodných prípadov (uzavretých poisťných zmlúv) a ich finančné sledovanie. V prvom rade dať dátam štruktúru, aby boli prehľadné a rýchlo dostupné pre interné použitie, ako aj pre vytváranie priebežných a uzávierkových správ a prehľadov pre predstavenstvo spoločnosti.

Na trhu existuje veľa produktov zaoberajúcich sa Business Intelligence, no spoločnosť BD FINANCE a.s. sa rozhodla pre vytvorenie vlastnej databázovej aplikácie. Vzhľadom na veľkosť firmy a ich požiadavky na túto aplikáciu a, samozrejme, aj z finančného hľadiska, je v danom prípade ich rozhodnutie tou správnou voľbou. Uspokojiť potreby spoločnosti by mala databázová aplikácia, ktorá bude presne podľa ich predstáv. Ako sa hovorí, „bude im užitá na mieru“. Na vytvorenie tejto aplikácie by mal postačovať program Microsoft Access.

Databázový program Microsoft Access dáva široké možnosti pre tvorbu a správu najrôznejších databázových súborov. Prepracovaný systém tabuliek, formulárov, dotazov a zostáv predstavuje veľmi mocný nástroj pre organizáciu dát v počítači. Mojm cieľom, však nebude len vytvorenie hore uvedených databázových objektov, ale aj vytvorenie celistvej aplikácie z týchto objektov. To uľahčí, spríjemní a sprehľadní zamestnancom prácu s dátami.

1 Teoretické východiská tvorby databázových aplikácií

1.1 Relačná databáza

Relačné databázy sú databázy, ktoré vyhovujú relačnému modelu dát (RMD). RMD je založený na matematickej teórii množín a predikátovej logike a definuje spôsob reprezentácie dát, spôsob ich ochrany (integritné obmedzenia) a možné operácie nad dátami. RMD navrhol a jeho pravidlá publikoval Dr. E. F. Codd (zamestnanec IBM) v r.1970 v článku: „A Relation Model of Data for Large Shared Databanks“. Okrem základných definícií, vychádzajúcich z teórie množín a predikátovej logiky, boli v článku obsiahnuté tieto myšlienky [11]:

- RMD oddeľuje dáta, ktoré sú chápané ako relácie od ich implementácie.
- Symetrický prístup k dátam. Pri manipulácii s dátami nás nezaujímajú prístupové metódy k dátam v reláciách.
- Pre prácu s dátami máme k dispozícii relačný kalkul a algebru (matematické aparáty, ktorými je možné popísať sémantiku (význam konštrukcií) relačných jazykov).
- Pre obmedzenie redundancie dát máme k dispozícii pojmy pre normalizáciu relácií, čo znamená vhodne navrhovať databázové štruktúry.

Základným kameňom RMD je databázová relácia (množina) obsahujúca dáta, ktorá sa od matematickej relácie líši. Je vybavená pomocnou štruktúrou, ktorej sa hovorí schéma relácie. Táto schéma obsahuje meno relácie, mená atribútov (stĺpcov) a popisuje domény (integritné obmedzenia). Každá bunka dát musí obsahovať iba skalárne hodnoty. [11]

Relácie je možné chápať ako tabuľku dát usporiadanú do stĺpcov (atribút + doména) a riadkov (n-tíc – n rozmerný vektor). Databázová relácie je tabuľka, pohľad, výsledok dotazu, čo nám dáva možnosť pracovať s výsledkom dotazov rovnako ako s tabuľkou. Vzhľadom na to, že relácia je množina, ktorá nesmie obsahovať duplicitné prvky a nie je usporiadaná inak ako do stĺpcov a riadkov, neexistuje prvý, druhý alebo n-tý riadok. Riadky v relácii nemajú špecifické poradie, teda ani nie sú dosiahnuteľné číslom riadku, musí existovať nejaká konštrukcia, ktorá nám umožní adresovať jednotlivé riadky. Tato konštrukcia sa nazýva primárny kľúč. [11]

Primárny kľúč je atribút alebo sústava atribútov, ktorých hodnoty tvoria jednoznačnú identifikáciu riadkov relácie. Každá relácia musí obsahovať primárny kľúč, v najhoršom prípade sú ním všetky atribúty. Každý atribút, ktorý je súčasťou kľúča, sa nazýva kľúčový, ostatné sú neklúčové. [11]

Databázový systém (DBS – Database System) je tvorený bázou dát, teda databázou (DB – Database, samotné dáta) a systémom riadenia báze dát SRBD (DBMS – Database Management System). Čo je sada programov, ktorá ukladá dáta a svojím špecifickým spôsobom určuje spôsob a techniky prístupu k dátam. [11]

Relačná databáza je implementácia modelov reálneho sveta, vytvorená podľa pravidiel RMD. Relation Database Management System je DBMS, ktorý by mal spĺňať 12 pravidiel, ktoré publikoval pán Codd v článku, ktorý vyšiel v Computerworlde (1985). Týchto 12 pravidiel vychádza z Coddovej teoretickej práce na relačnom modeli a vyjadruje skôr ideálne ciele, ktoré by mal dosiahnuť relačný DBMS [11]:

- 1) Pravidlo informácie: Všetky informácie v relačnej databáze sa na logickej úrovni reprezentujú explicitne hodnotami v tabuľkách.
- 2) Pravidlo zaručeného prístupu: Musí byť zaistené, aby každý údaj v relačnej databáze bol logicky prístupný použitím názvu tabuľky, hodnoty primárneho kľúča a názvu stĺpca.
- 3) Systematické ošetrenie prázdnych hodnôt: Prázdne hodnoty (nie nuly, či prázdne reťazce) sú systematicky plne podporované RDBMS pre reprezentáciu chýbajúcich informácií a neplatných informácií nezávisle na dátovom type. (Typicky riešené pomocou hodnoty NULL).
- 4) Popis štruktúry založený na relačnom modeli: Popis databáze sa na logickej úrovni reprezentuje rovnako ako bežné dáta , tzn. v reláciách, na ktoré sa môžu oprávnení užívatelia dotazovať rovnako ako na akúkoľvek inú reláciu.
- 5) Pravidlo komplexného dátového jazyka: Relačné systémy môžu podporovať viac jazykov a režimov prístupu, ale musí existovať minimálne jeden jazyk, ktorého príkazy sú vyjadriteľné nejakou dobre definovanou syntaxou ako reťazce znakov, ktorý podporuje:

- a) definíciu dát
- b) definíciu pohľadov
- c) manipuláciu s dátami
- d) obmedzenie integrity
- e) autorizáciu
- f) vymedzenie transakcie

V súčasnosti používa väčšina DBMS jazyk SQL.

- 6) Aktualizácia pohľadu: Všetky aktualizovateľné pohľady je možné aktualizovať systémovo.
- 7) Vysoko úrovňová manipulácia s dátami: Spracovanie základní či odvodenej relácie ako jediný operand sa aplikuje ako na vyhľadávanie, tak vloženie a zmenu dát.
- 8) Fyzická dátová nezávislosť: Aplikácia a terminály zostávajú logicky nedotknuté zmenami v reprezentácii úložiska alebo v prístupových metódach.
- 9) Logická dátová nezávislosť: Aplikácia a terminály sú logicky nedotknuté, pokiaľ sú v tabuľkách vykonané zmeny v uchovaní informácií.
- 10) Nezávislosť integrity: Integritné obmedzenia musia byť definovateľné v dátovom jazyku v databáze, nie v aplikácii.
- 11) Distribučná nezávislosť: Databázový jazyk musí byť schopný manipulovať s dátami umiestnenými na inom počítačovom systéme.
- 12) Pravidlo nenarušenia: Pokiaľ je v systéme viac jazykov, žiadny z nich nesmie mať možnosť manipulovať s dátami v rozpore s integritnými obmedzeniami.

Prakticky žiadny zo súčasných systémov nespĺňa úplne dokonale všetky pravidlá a mnoho programátorov databázových aplikácií nevyužíva možnosti databáze. (Týka sa to hlavne bodu 10).

1.1.1 Normalizácia

Normalizácia je odstránenie redundantných (opakujúcich) sa dát, obmedzenie zložitosti (rozložením zložitej relácie na dvojrozmerné tabuľky) a zabránenie takzvaným aktualizáčnym anomáliám. Čo by malo viesť k databáze prehľadnejšej, rozšíriteľnejšej a výkonnejšej. [10]

Normalizácia by mala viesť k vzniku tabuliek, ktoré je možné jednoducho udržiavať a efektívne sa na ne dotazovať. Normalizovaná schéma musí zachovať všetky

závislosti pôvodných schém a relácia musí zachovať pôvodné dáta, čo znamená, že sa musíme pomocou prirodzeného spojenia dostať k pôvodným dátam. [10]

Normalizovať je určite potreba a čím zložitejšia databáza a čím viac dát, tým viac je potreba normalizovať. Aj tu však platí všetko s mierou.

1.1.2 Integritné obmedzenie

Integrita databáze je inak povedané konzistencia. Relačný model dát špecifikuje štruktúru dát v databáze, ale k tomu, aby sme mohli používať databázu ako zdroj dát, je nutné zaistiť, aby sa do nej dostali len dáta, ktoré tam patria a nestratili sa dáta, ktoré by sa nemali. Tiež je potrebné mať k tomu určité mechanizmy. Týmito mechanizmami sú integritné obmedzenia (IO). Databáza, respektíve dáta, sú konzistentné, pokiaľ vyhovujú IO. To znamená, že so žiadnou úpravou dát (úpravou, zmazaním) nestratila alebo nepoškodila dáta, alebo že v DB nie sú dáta, ktoré tam nemajú čo robiť. Preto existujú integritné obmedzenia, ktoré majú podobným nehodám zabrániť. Vzhľadom na to, že k porušeniu integrity databázy môže dôjsť niekoľkými spôsobmi, rozoznávame niekoľko druhov integritných obmedzení [9]:

- **Entitné**

Toto IO všetci používame, pretože je v relačnom modeli povinné, ide o špecifikáciu primárneho kľúča tabuľky. Primárny kľúč je atribút, či minimálny zoznam atribútov, ktoré jednoznačne určujú n-ticu (riadok) relácie. Minimálne v zmysle, že z nej nie je možné odobrať žiadny atribút bez toho, aby sa stratila jedinečnosť. Toto obmedzenie implementuje každá relačná databáza a pri pokuse o vloženie dát, ktoré nevyhovujú tomuto obmedzeniu, dôjde ku chybe.

- **Doménové**

Doménová integrita znamená, že na úrovni stĺpcov definujeme obmedzenie na určitý dátový typ, prípadne obmedzenie rozsahu hodnôt.

- **Referenčné**

Referenčná integrita je už medzitableková záležitosť. Definuje vzťah dvoch tabuliek, a to pomocou cudzích kľúčov (Foreign Key). Tabuľky, ktorých sa tento vzťah týka, bývajú anglicky označované ako master - detail alebo parent - dependent. Cudzí kľúč v relácii určuje atribút (skupinu atribútov), ktoré majú buď hodnotu NULL alebo hodnotu primárneho kľúča niektorého riadku nadradenej tabuľky.

Aktívna referenčná integrita definuje, čo má databázový stroj vykonať, pokiaľ by malo dôjsť k porušeniu referenčnej integrity. Napríklad pri zmazaní nadradeného záznamu je možné u väčšiny moderných databázových systémov definovať akciu, ktorá sa má vykonať (napr. zmazať do kaskády, atď.).

1.2 Microsoft Access

Access je súčasťou sady programov Office od spoločnosti Microsoft. MS Access patrí medzi relačné DBMS. Umožňuje spravovať informácie v jedinom súbore. V rámci tohto súboru sa používajú nasledujúce základné objekty:

1.2.1 Tabuľky

Tabuľka je jeden zo základných objektov, s ktorými môžeme v MS Access pracovať. Zo všetkých základných typov databázových objektov slúžia k ukladaniu dát iba tabuľky. Ostatné sa používajú k správe, manipuláciám, analýze, získavaniu, zobrazovaniu alebo publikovaniu informácií z tabuliek. Databáza v MS Access môže obsahovať tisíce tabuliek a počet záznamov, ktoré môže každá tabuľka obsahovať, je limitovaný prevažne tým, koľko miesta máte na svojom pevnom disku. [3, s. 221, 224]

Každý objekt MS Access má dve alebo viac zobrazení. Pri tabuľkách sú dve najpoužívanejšie: zobrazenie dátového listu, v ktorom môžeme prezerať a upravovať dáta tabuľky, a návrhové zobrazenie, v ktorom môžeme prezerať a upravovať štruktúru tabuľky. Pokiaľ majú dve tabuľky spoločné jedno alebo viac polí, môžeme do jedného dátového listu vložiť dátový list inej tabuľky. Vložené dátové listy (vnorené dátové listy) umožňujú naraz prehliadať dáta z viacerých tabuliek. [3, s. 225, 226]

Keď vytvoríme tabuľky a špecifikujeme ich polia pomocou sprievodcu, nastaví sprievodca u každého poľa skupinu vlastností (viď obr. 1.1). Tieto vlastnosti určujú, aké dáta je možné do poľa zadávať a ako budú dáta vyzeráť na obrazovke. Štruktúru tabuľky a vlastnosti polí môžeme doladiť v návrhovom zobrazení. Okno návrhového zobrazenia tabuľky obsahuje v hornej časti zoznam polí tabuľky. Stĺpec „Názov poľa“ obsahuje názvy, ktoré sme špecifikovali pri vytváraní tabuľky (v názvoch nie sú medzery). Stĺpec „Dátový typ“ určuje typ dát, ktoré môže pole obsahovať. Do stĺpca „Popis“ je možné priradiť vysvetlenie, čo pole znamená. Potom priradíme primárny kľúč. Ten sa priraduje na jedno alebo viacero polí. Hodnota tohto kľúča jednoznačne identifikuje záznam. [3, s.256, 258, 259]

Obecné	Vyhledávání
Velikost pole	4
Formát	
Vstupní maska	
Titulek	Pravděpodobnost získání klienta
Výchozí hodnota	
Ověřovací pravidlo	
Ověřovací text	
Je nutno zadat	ne
Povolit nulovou délku	ano
Indexovat	ano (duplicita povolena)
Komprese kódu Unicode	ne
IME Mode	No Control
IME Sentence Mode	No Conversion

Obr. 1.1 Vlastnosti pole v návrhovom zobrazení tabuľky pre dátový typ text

Nakoniec môžeme vytvoriť relačné väzby medzi tabuľkami. To väčšinou znamená, že vytvoríme reláciu medzi poľom s primárnym kľúčom jednej tabuľky a poľom s cudzím kľúčom druhej tabuľky (relácia v Microsoft Access neznamená tabuľka, ale vzťah medzi tabuľkami). Tieto polia musia z pravidla byť rovnakého dátového typu a mať rovnakú veľkosť. Rozoznávame tri typy relácií 1 - 1, 1 - N a M - N.

1.2.2 Dotazy

Jeden zo spôsobov vyhľadávania informácií v databáze MS Access je vytváranie dotazov. Dotazy sa používajú k nájdeniu špecifických informácií, ktoré potom môžeme rôznymi spôsobmi prehliadať, meniť či analyzovať. Výsledky dotazov môžu slúžiť tiež ako podkladové dáta pre formuláre a zostavy. [3, s. 229]

Dotaz je vo svojej podstate otázka. Aby bol MS Access schopný na tieto otázky odpovedať, musíme svoje otázky klásť veľmi špecifickým spôsobom. Každému typu otázky zodpovedá určitý typ dotazu. Primárne typy dotazov sú [3, s.229, 316, 317]:

- **Výberový dotaz** získava dáta z jednej alebo viacerých tabuliek a zobrazí výsledky v dátovom liste. Výberový dotaz sa dá tiež využiť pre zoskupovanie záznamov a výpočet rôznych štatistík ako sú súčty, priemery, počty hodnôt a iné typy súhrnov (tzv. agregáčné dotazy). Záznamy pochádzajúce z jednej alebo viacerých tabuliek prepojených reláciami môžeme v dátovom liste aktualizáčného dotazu aktualizovať.
- **Parametrický dotaz** je dotaz, ktorý vás požiada o nejaké informácie, ktoré potom použije pri spracovaní dotazu (napr. dátum od - do).
- **Krížový dotaz** vypočítava dáta a mení ich štruktúru pre potreby ďalších analýz. Vie spočítať súčty, priemery, četnosti a iné druhy súhrnov dát, ktoré sú zoskupené podľa

dvoch typov informácií – jednej pozdĺž ľavej strany dátového listu, druhej pozdĺž hornej strany. Výsledky dotazu sa zobrazia v bunkách, ktoré sú v priesečníkoch riadkov a stĺpcov. (Je to obdoba kontingenčnej tabuľky)

Ďalší typ dotazov sú dotazy akčné:

- **Akčné dotazy** vykonávajú hromadné akcie nad záznamami ako jedinú operáciu (transakcia). Dostupné sú štyri druhy akčných dotazov: odstraňovacie (DELETE), aktualizácie (UPDATE), pridávacie (INSERT) a vytváracie (SELECT INTO).

Pre tvorbu dotazov poskytuje MS Access sprievodcu, pomocou ktorého rýchlo zostrojíme bežnejšie typy dotazov. Zložitejšie dotazy sa vytvárajú v návrhovej mriežke návrhového zobrazenia dotazu alebo si môžeme dotaz nadefinovať ručne pomocou jazyka SQL.

1.2.3 Formuláre

Formulár je vo svojej podstate okno, do ktorého sa umiestňujú ovládacie prvky, ktoré buď poskytujú užívateľovi nejaké informácie, alebo im umožňujú informácie zadávať. MS Access poskytuje súpravu nástrojov, na ktorej nájdeme mnoho štandardných ovládacích prvkov Windows, ako sú popisy, textové pole, polia so zoznamom, prepínače či zaškrŕavacie políčka. Formulár môže tiež obsahovať podformuláre, ktoré sú napojené na iné tabuľky ako hlavný formulár. [3, s. 233, 234]

Formuláre sa využívajú pre editovanie záznamov podkladových tabuliek alebo pre zadávanie nových záznamov. Podobne ako je to u tabuliek a dotazov aj formulár môžeme zobraziť niekoľkými spôsobmi. Najbežnejšie sú: formulárové zobrazenie, v ktorom sa obvykle zadávajú dáta, a návrhové zobrazenie, v ktorom pracujeme s prvkami formuláru, keď potrebujeme zdokonaľiť vzhľad formuláru a jeho chovanie. [3, s. 233]

1.2.4 Zostavy

Zostavy tiež slúžia k zobrazovaniu informácií z tabuliek, ale účelom je predovšetkým, aby boli dáta pekne naformátované a usporiadané a aby sa dalo jednoducho prehliadať buď na obrazovke počítača, alebo vytlačené na papieri. Zostava môže obsahovať údaje, ktoré pochádzajú z niekoľkých tabuliek a dotazov, môže zahŕňať hodnoty vypočítané z informácií uložených v databáze a obvykle tiež obsahuje prvky štábnej kultúry, ako sú záhlavie a zapätie, titulky a nadpisy. [3, s.237]

Zostavy majú tiež viacero zobrazení. Návrhové zobrazenie slúži predovšetkým k úpravám rozvrhnutia a návrhu zostavy. V zobrazení náhľad vidíme zostavu tak, ako bude vyzerat' vytlačená.

1.2.5 Ostatné objekty

Medzi ostatné objekty patria stránky, makrá a moduly VBA.

1.3 Vytvorenie aplikácie v Microsoft Access

Databázová aplikácia je rovnako ako databáza tvorená databázovými objektmi. Tými sú tabuľky, formuláre, dotazy, zostavy, makrá a moduly. Hlavným rozdielom medzi databázou a aplikáciou je v tom, že v aplikácii tvoria databázové objekty jeden ucelený systém. Prejaví sa to napríklad tým, že aplikácia sama riadi beh súvisiacich úloh (napr. vyhľadávanie niektorých dát apod.) a užívateľ nie je pri svojej práci zahltený množstvom okrajových problémov súvisiacich s organizáciou databáze. [2]

Databázové objekty sa navyše delia na objekty, ktoré užívateľ priamo vidí a používa (zostavy a formuláre) a na podporné objekty. Podporné objekty sú teda moduly, makrá, dotazy a dokonca aj tabuľky. V správne vytvorenej aplikácii už nie je potrebné (pretože to nie je efektívne) meniť dáta priamo v zobrazení dátového listu tabuliek. Vždy sa otvorí príslušný formulár, v ktorom je úprava dát jednoduchšia a prehľadnejšia. [2]

Všetky objekty majú v aplikáciách vlastnosti, ktorých nastavením docielime správnu funkciu aplikácie. [2]

Zdatným pomocníkom pri tvorbe aplikácie je možnosť definovania vlastnej odozvy na udalosti všetkých typov. Klasickým príkladom takej odozvy je spustenie makra (sekvencia akcií Microsoft Access) alebo udalostnej procedúry, čo je nami naprogramovaná procedúra programovacieho jazyka Visual Basic for Application (VBA). Niektoré udalosti v behu aplikácie sú ošetrené štandardne. Napríklad zmena dát v poliach formulára sa premietne do tabuliek, ktoré (aj keď sú iba podporné objekty) stále uchovávajú najdôležitejšiu časť dát aplikácie. [2]

Aplikáciu môžeme vytvoriť buď samostatne alebo pomocou sprievodcu. V tom prípade si môžeme vybrať z mnohých štandardných typov aplikácií, ktoré neskôr upravíme a preprogramujeme tak, aby čo najlepšie zodpovedali našim požiadavkám. [2]

1.3.1 Vytvorenie aplikácie bez pomoci sprievodcu

Vytvorenie aplikácie bez pomoci sprievodcu je síce časovo náročnejšie, zato však môžeme vytvoriť takú aplikáciu, ktorá bude presne vyhovovať našim predstavám a požiadavkám. Pokiaľ vytvárame aplikáciu bez pomoci sprievodcu, musíme si najprv dobre rozmyslieť, aký typ dát budeme do aplikácie ukladať, a podľa toho navrhnuť systém tabuliek a relácií medzi nimi. Táto časť návrhu aplikácie je vôbec najdôležitejšia, pretože pri zlom návrhu aplikácie nie je možné využiť všetky prostriedky, ktoré systém Microsoft Access ponúka. Neskoršia zmena štruktúry ukladania dát je síce možná, ale zdĺhavá a často nepríjemná. Pokiaľ sa takejto práci chceme vyhnúť, musíme venovať dostatok času práve návrhu aplikácie. [2]

Pokiaľ máme vytvorené všetky príslušné tabuľky (môžeme k tomu využiť sprievodcu tabuľkou a drobné detaily spresniť v návrhovom zobrazení tabuľky), je na čase vyrobiť formuláre prislúchajúce k jednotlivým tabuľkám a prispôbiť ich vzhľad pomocou návrhového zobrazenia formuláru. Táto činnosť je veľmi dôležitá, pretože z hľadiska užívateľa, ktorý bude našu databázovú aplikáciu používať, tvoria formuláre prakticky celú aplikáciu a všetky rozhrania. [2]

Ďalším krokom je vytvorenie makier a modulov VBA potrebných pre správnu funkciu formulárov. Vhodnú formu predstavujú udalostné procedúry. Tieto procedúry a makrá pripojíme k príslušným príkazovým tlačidlám a udalostným vlastnostiam. [2]

Je vhodné funkciu každého formuláru (prípadne iného databázového objektu) hneď po vytvorení podrobne vyskúšať, aby už nebolo nutné sa k nemu vracieť. Funkcie, ktoré nie je možné vyskúšať okamžite (napr. komunikácia s objektom, ktorý ešte nie je vytvorený) je nutné odskúšať neskôr s rovnakou presnosťou. [2]

Pri vytváraní aplikácie musíme myslieť na to, že náš produkt pravdepodobne nebude slúžiť iba nám, ale bude ho využívať široké spektrum rôzne skúsených užívateľov. Z tohto dôvodu je potrebné, aby systém ovládania takejto aplikácie bol pokiaľ možno čo najjednoduchší, ale aj napriek tomu spĺňal požadované nároky na komfort a efektivitu práce. Posledným krokom pri vytváraní aplikácie by potom mohlo byť vytvorenie hlavného formuláru, ktorý sa zobrazí po spustení aplikácie, prípadne ďalších databázových objektov (napr. zostáv). [2]

Hlavný formulár je prvým objektom, s ktorým užívateľ príde do kontaktu po spustení aplikácie. Mal by teda byť dostatočne jasný všetkým užívateľom a umožňovať

ľahký prístup ku všetkým nástrojom aplikácie, ktoré sa používajú najčastejšie (napr. vyhodnocovanie predaja, tlačenie zostáv, vyvolanie formuláru pre zadanie nových dát, atď.). Existujú dva typy najpoužívanejších typov úvodných formulárov [2]:

- Úvodný formulár je koncipovaný ako prepínací panel, ktorý obsahuje iba príkazové tlačidlá, ktorých stlačením sa užívateľ presunie do konkrétnej časti aplikácie využívajúcej určitý databázový objekt. Do takéhoto formuláru je vhodné zahrnúť aj tlačidlo typu „Zavrieť aplikáciu“.
- Pokiaľ je celá aplikácia monotematicky zameraná na jedinú hlavnú činnosť, je vhodné využiť príslušný formulár k tejto činnosti ako úvodný.

Pre vykonávanie akcií v rámci jedného formuláru alebo pre otváranie ostatných formulárov a iných databázových objektov je možné definovať niekoľko nástrojov. Sú to predovšetkým príkazové tlačidlá pracovnej plochy, príkazy horizontálneho menu, definované kombinácie kláves (tzv. hot keys) a panely nástrojov. Niektoré tieto ovládacie prvky definuje sám sprievodca pri vytváraní príslušného databázového objektu (väčšinou formuláru), ale všetky je možné pretvoriť alebo nanovo pridať v návrhovom zobrazení príslušného databázového objektu. [2]

- Príkazové tlačidlo sa používa k vykonaniu jedného úkonu. Je možné ho umiestniť do databázového objektu v jeho návrhovom zobrazení kliknutím na príslušný symbol súpravy nástrojov. Tlačidlo umiestnime myšou na požadované miesto pracovnej plochy a nastavíme jeho vlastnosti. Najdôležitejšia je v tomto prípade udalostná vlastnosť „Pri kliknutí“. K tejto udalosti priradíme makro, akciu alebo udalostnú procedúru, ktorá má byť vyvolaná pri stlačení tohto tlačidla vo formulári.
- Ďalším štandardným nástrojom pre ovládanie aplikácií sú príkazy horizontálneho menu. Horizontálne menu, ktoré sami vytvárame, sa delí na globálne a lokálne. Globálne menu sa zobrazí na pracovnej ploche vo všetkých dialógových oknách okrem tých formulárov a zostáv, ktoré majú svoje lokálne menu pripojené práve k tomuto databázovému objektu. V takomto prípade lokálne horizontálne menu prekryje menu globálne.
- Často používané príkazy z horizontálneho menu je možné doplniť do panelu nástrojov vo forme príkazových tlačidiel alebo grafických ikon. V dialógovom okne „Panely nástrojov“ môžeme zvoliť, ktorý panel nástrojov chceme zmeniť, prípadne vytvoriť panel úplne nový. Pri úprave panelu nástrojov môžeme zmeniť vzhľad jednotlivých

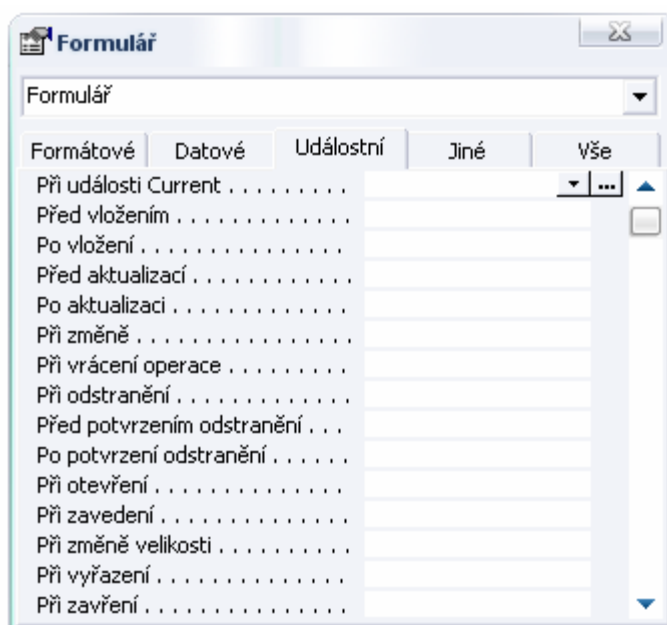
ikon alebo pridať ikony nové. Pomocou nových ikon je možné otvárať rôzne databázové objekty aplikácie alebo spúšťať makro.

- Veľmi elegantný spôsob ovládania aplikácie je pomocou klávesových skratiek. Tieto fungujú nezávisle na tom, ktorý databázový objekt je práve aktívny. K definovaniu týchto klávesových skratiek slúži špeciálne makro s názvom „Automatické klávesy“.

1.4 Visual Basic for Application

Programovací jazyk Visual Basic for Application (VBA), ktorý je vstavanou súčasťou databázového systému Microsoft Access, poslúži predovšetkým tam, kde požiadavky na výkon a ovládanie aplikácie sú vyššie ako tie, ktoré poskytujú štandardné nástroje Microsoft Access. VBA sa, podobne ako makrá, používa predovšetkým k previazaniu databázových objektov v jeden integrovaný celok. Správne naprogramované procedúry programovacieho jazyka však vykazujú v porovnaní s makrami vyšší výkon. VBA tiež predstavuje flexibilnejší nástroj pre riadenie aplikácií a výsledný programovací kód tak môže byť užívateľovi „ušíty priamo na mieru“. Oproti tomu je používanie makier, obzvlášť pre niekoho, kto nikdy neprogramoval, jednoduchšie a zrozumiteľnejšie. [2]

Aj pre stredne skúsených programátorov však VBA predstavuje lákavú alternatívu pri vytváraní výkonných aplikácií. [2]



Obr. 1.2 Udalostné vlastnosti formuláru

Programové riadenie databázových aplikácií je príkladom programu riadeného udalosťami. Programátor vytvorí iba súbor procedúr, ktoré majú byť spustené v príslušnom formulári, zostave pri výskyte určitej udalosti (napr. pri kliknutí na príkazové tlačidlo, po aktualizácii textového poľa, atď.)(viď obr. 1.2). V systéme Microsoft Access nie je potrebné písať program, ktorý by kontroloval, ktorá udalosť práve nastala. Toto robia databázové objekty automaticky. Pri výskyte každej udalosti sa potom vykoná procedúra alebo makro, ktoré je k danej udalosti pripojené. [2]

1.4.1 Výhody programového kódu VBA

- Aplikácie s programovým kódom VBA sa udržujú a rozširujú jednoduchšie ako aplikácie založené na ovládaní pomocou makier. Makra tvoria samostatné databázové objekty, zatiaľ čo programový kód je vstavaný do jednotlivých formulárov, zostáv, atď. Pri akejkolvek manipulácii s týmito objektmi (napr. kopírovanie do inej aplikácie) sa tak príslušný programový kód automaticky prenáša so svojím objektom, čo uľahčuje ďalšie úpravy aplikácií.
- Pomocou VBA je možné ošetriť možné chyby, ktoré vzniknú pri behu aplikácie a nahradiť často záhadné chybové hlásenia systému vlastnými správami. Takéto ošetrenie behu aplikácie je zvlášť výhodné pri viac-užívateľských inštaláciách.
- Programový kód VBA je schopný manipulovať so všetkými objektmi aplikácie, pokiaľ je to potrebné.
- Procedúra VBA je schopná komunikovať aj so súbormi mimo systém Microsoft Access.
- Ďalšiu výhodu predstavuje možnosť meniť hodnoty argumentov jednotlivých funkcií VBA. Pri použití makier túto možnosť nemáme. Tam musíme definovať všetky argumenty pri definícii makra.
- VBA umožňuje manipuláciu s jednotlivými záznamami a nielen so sadou záznamov.

2 Analýza súčasného stavu

Ako už bolo uvedené v úvode, spoločnosť BD FINANCE a.s. je pomerne mladá, nakoľko sa jedná o novovzniknutú spoločnosť. V súčasnosti sa v spoločnosti nepoužíva žiadny nástroj Business Intelligence ani žiadna databázová aplikácia.

Zamestnanci využívajú iba program Microsoft Excel na vytvorenie jednoduchých tabuliek, ktoré slúžia na evidovanie poistných zmlúv a finančné sledovanie poistných zmlúv.

Podnik	č.smlouvy	Pojišťovna	Druh pojištění	Správce
BBBB	7720215470	Alianz	majetek a odpovědnost	J.Novák
AAAA	10086013	ČPP	odpovědnost za škodu	J.Novák
CCC	899-12564-10	ČP	závažné havárie	J.Novák
CCC	706-57220-10	ČP	majetek	J.Novák
DDD	2116286317	Alianz	majetek a odpovědnost	J.Novák

Tab. 2.1 Evidencia poistných zmlúv

Zdroj: [12], autorsky upravené

Začátek pojištění	Konec pojištění	Roční pojistné	% Provize	Roční provize v Kč
1.8.2007	31.7.2008	433 204,00	15%	64 980,60
11.6.2007	10.6.2008	35 700,00	0%	0,00
27.9.2007	26.9.2008	1 090 000,00	9%	98 100,00
1.10.2007	30.9.2010	1 450 136,00	18%	261 024,48
14.9.2007	září neurčito	486 815,00	18%	87 626,70

Tab. 2.2 Evidencia poistných zmlúv

Zdroj: [12], autorsky upravené

Ako vidieť v tabuľkách tab. 2.1, 2.2 a 2.3, ktoré znázorňujú súčasné riešenie evidencie obchodných prípadov a ich finančné sledovanie pomocou programu MS Excel, je tento spôsob nevyhovujúci a to najmä z dôvodu neprehľadnosti dát. Všetky záznamy sú vedené v jednej veľkej tabuľke, čo so sebou prináša rôzne problémy.

Prvé problémy môžu nastať už pri zadávaní dát, kde môže ľahko dôjsť k chybám, ktoré sa neskôr prejaví v nepresných výsledkoch a iných anomáliách. Takisto pri ich následnom vyhľadávaní, filtrovaní a zostavovaní výstupov, ktoré si navyše vyžadujú od užívateľov určité schopnosti a znalosti nie úplne triviálneho charakteru.

Dôležitým faktorom je samozrejme časová náročnosť. V podnikaní ešte stále platí pravidlo „čas sú peniaze“. Poist'ovací maklér nemôže stráviť polovicu svojej pracovnej doby zdĺhavým vyhľadávaním a vyplňaním údajov, ktoré už navyše niekoľkokrát predtým zadával. Rýchlosť a dostupnosť informácií znamená pre firmu aj veľkú konkurenčnú výhodu na trhu.

So súčasným riešením nie sú spokojní ani samotní zamestnanci spoločnosti BD FINANCE a.s..

<u>2007</u>				
Srpen PP	Srpen SP	Září PP	Září SP	Říjen PP
		16 245,00	16 245,00	
				98 100,00
				65 256,00
		21 906,68		
2 013,34				
				686,56
		3 262,50		
1 502,20				
				1 657,38
		2 808,12		
				627,62
				9 750,00
				1 262,10
3 515,54	0,00	44 222,30	16 245,00	177 339,66

Tab. 2.3 Finančné sledovanie

Zdroj: [12], autorsky upravené

3 Návrh a zavedenie databázovej aplikácie

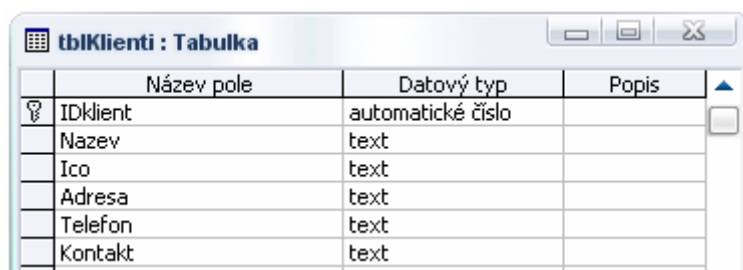
Databázová aplikácia, ktorú vytváram, sa svojim obsahom nepribližuje žiadnej z existujúcich šablón, preto nie je možné využiť sprievodcu a následne už aplikáciu len upraviť a doprogramovať. Rozhodol som sa preto, že databázovú aplikáciu pre firmu BD Finance a.s. od základov navrhнем sám.

3.1 Dátová štruktúra

Správne navrhnutá dátová základňa je pri tvorbe databázovej aplikácie veľmi dôležitá, preto som bol viackrát osobne na konzultáciách so zákazníkom, na ktorých sme sa dohodli na štruktúre dát a na tom, aké dáta majú tabuľky (respektíve formuláre a zostavy) obsahovať.

Zákazník si prial mať rozdelenú evidenciu obchodných prípadov na životnú a neživotnú časť. K rozdeleniu som pristúpil už v návrhu dátovej štruktúry.

3.1.1 Popis a štruktúra tabuliek



	Název pole	Datový typ	Popis
IDklient	IDklient	automatické číslo	
Nazev	Nazev	text	
Ico	Ico	text	
Adresa	Adresa	text	
Telefon	Telefon	text	
Kontakt	Kontakt	text	

Obr. 3.1 Tabuľka tblKlienti (neživotný)

Táto tabuľka obsahuje údaje o klientoch spoločnosti. Konkrétne o klientoch neživotných (tzn. právnické osoby). Pole IDklient je nastavené ako primárny kľúč a jeho hodnotu nebude zadávať užívateľ, ale bude generovaná automaticky. Toto zabezpečuje dátový typ automatické číslo. Ostatné polia sú dátového typu text, rôznej dĺžky. Na poli Ico je nadefinovaná vstupná maska (00000000;0) a dané pole je indexované bez duplicity, aby som zabránil chybám a viacnásobnému vkladaniu tých istých klientov.

Podobne je to v tabuľke tblKlientiZ (určená pre evidenciu fyzických osôb)(viď obr. 3.2), kde je pole RodCislo (rodné číslo) tiež indexované bez duplicity a so vstupnou maskou (000000"/"0000;0).

	Název pole	Datový typ	Popis
PK	IDklientZ	automatické číslo	
	Jmeno	text	
	Primeni	text	
	Telefon	text	
	RodCislo	text	
	PoznamkaKlientZ	memo	

Obr. 3.2 Tabuľka tblKlientiZ (životný)

	Název pole	Datový typ	Popis
PK	IDmakler	automatické číslo	
	Jmeno	text	
	Primeni	text	

Obr. 3.3 Tabuľka tblMakleri

Ďalšou v poradí je tabuľka tblMakleri, v ktorej sú údaje o makléroch pracujúcich v spoločnosti BD Finance a.s.. Pole IDmakler je primárny kľúč. Keďže táto tabuľka bude slúžiť len na identifikáciu maklérov, obsahuje okrem IDmakler už len polia Meno a Priezvisko. Pre evidenciu obchodných prípadov sú ostatné polia zbytočné.

	Název pole	Datový typ	Popis
PK	IDpojistovna	automatické číslo	
	Nazev	text	

Obr. 3.4 Tabuľka tblPojistovny

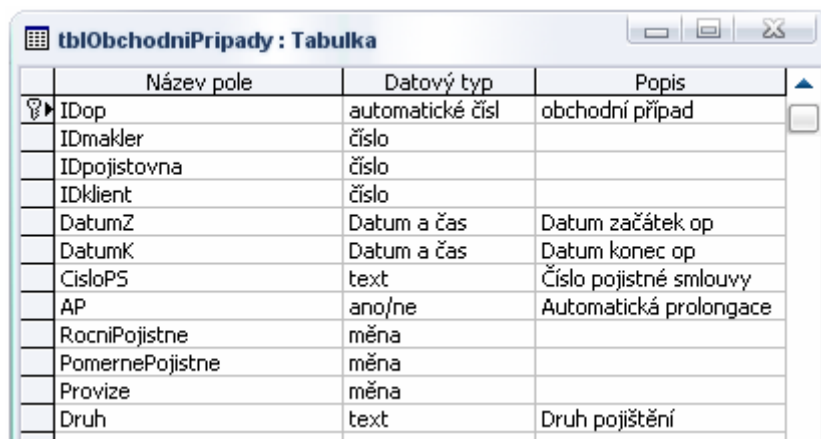
Podobné je to aj pri tabuľke tblPojistovny, ktorá by mala obsahovať len zoznam poisťovní, s ktorými spoločnosť BD Finance a.s. spolupracuje.

	Název pole	Datový typ	Popis
PK	IDakvizice	automatické číslo	
	IDklient	číslo	
	IDmakler	číslo	
	Pravdepodobnost	text	pravdepodobnosť získania klienta
	PrvniKontakt	Datum a čas	datum prvého kontaktu
	RocniProvize	měna	očekávaná ročná provízie
	DatumProvize	Datum a čas	datum prvej očakavanej provízie
	DatumPM	Datum a čas	datum plna moc

Obr. 3.5 Tabuľka tblAkvizice (neživotné)

Tabuľky tblAkvizice (neživotné) (viď obr. 3.5) aj tblAkviziceZ (životné) sú dorobené na žiadosť zákazníka, aby mohol evidovať obchodné prípady, ktoré ešte nie sú

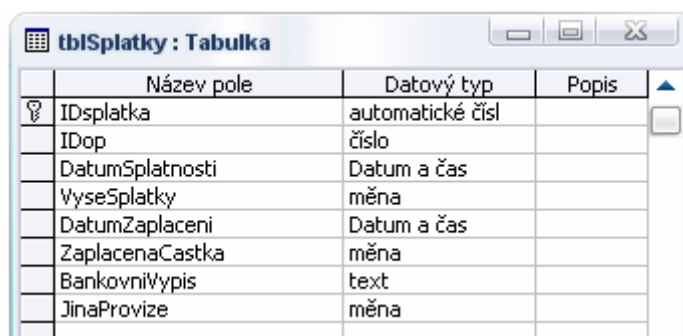
zazmluvnené (čaká sa na vypršanie starej zmluvy alebo sa ešte rokuje...). Pole s primárnym kľúčom je IDakvizice (IDakviziceZ u životných). Pole IDklient (IDklientZ u životných) je cudzím kľúčom k rovnomennému poľu v tabuľke tblKlienti (tblKlientiZ u životných) a pole IDmakler je cudzím kľúčom k rovnomennému poľu v tabuľke tblMakleri. Pole Pravdepodobnost má ovládací prvok pole so zoznamom a zdroj riadkov je zoznam hodnôt (10%, 20%...100%). Toto pole je obmedzené na zoznam, takže užívateľ si môže vybrať len niektorú z položiek v zozname.



	Název pole	Datový typ	Popis
?	IDop	automatické čísl	obchodní případ
	IDmakler	číslo	
	IDpojistovna	číslo	
	IDklient	číslo	
	DatumZ	Datum a čas	Datum začátek op
	DatumK	Datum a čas	Datum konec op
	CisloPS	text	Číslo pojistné smlouvy
	AP	ano/ne	Automatická prolongace
	RocniPojistne	měna	
	PomernePojistne	měna	
	Provize	měna	
	Druh	text	Druh pojištění

Obr. 3.6 Tabuľka Obchodné prípady (neživotné)

Tabuľky tblObchodniPripady (neživotné) a tblObchodniPripadyZ (životné) majú slúžiť na evidenciu už existujúcich obchodných prípadov (poistných zmlúv). Primárny kľúč je pole IDop (IDopZ u životných) a polia IDmakler, IDklient (IDklientZ u životných) sú cudzími kľúčmi, presne ako u akvizícií. Pole IDpojistovna je takisto cudzí kľúč k rovnomennému poľu v tabuľke tblPojistovny. Pole Druh je pole so zoznamom, obmedzené na zoznam a jeho prvky sú jednotlivé druhy poistenia (majetkové, havarijné, ...). Pole CisloPS je indexované bez duplicity, aby sa zabránilo vkladaniu rovnakých zmlúv viackrát. Polia s dátovým typom Datum a čas majú formát - datum(krátke). Pole Provize je formátu percentá s dvomi desatinnými miestami.



	Název pole	Datový typ	Popis
?	IDSplatka	automatické čísl	
	IDop	číslo	
	DatumSplatnosti	Datum a čas	
	VyseSplatky	měna	
	DatumZaplaceni	Datum a čas	
	ZaplacenaCastka	měna	
	BankovniVypis	text	
	JinaProvize	měna	

Obr. 3.7 Tabuľka Splátky (neživotné)

Tabuľky tblSplatky (neživotné) a tblSplatkyZ (životné) slúžia na finančné sledovanie obchodných prípadov, na zaznamenávanie splátok, ktoré sú platené v jednotlivých poistných zmluvách. Pole IDSplatka je primárny kľúč. Pole IDop je cudzí kľúč k rovnomennému poľu v tabuľke Obchodné prípady (IDopZ u životných). Polia s dátovým typom Datum a čas majú takisto formát datum(krátke) a pole JinaProvize má formát percento a dve desatinné miesta.

Název pole	Dátový typ	Popis
IDpoznamka	automatické číslo	
IDakvizice	číslo	
DatumPoznamka	Datum a čas	
Poznamka	memo	

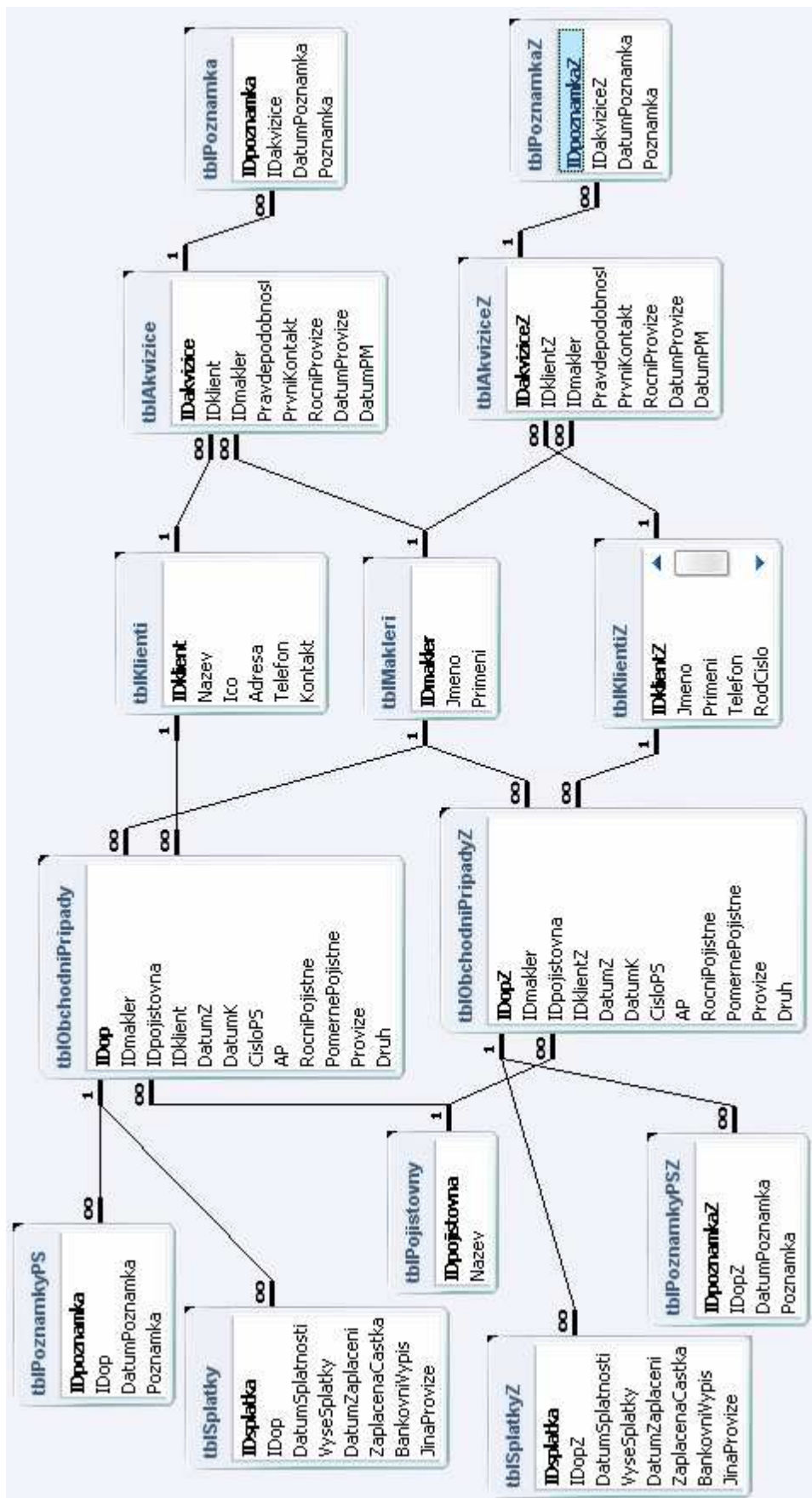
Název pole	Dátový typ	Popis
IDpoznamka	automatické číslo	
IDop	číslo	
DatumPoznamka	Datum a čas	
Poznamka	memo	

Obr. 3.8 Tabuľky Poznamka a PoznamkyPS

Ako posledné som vytvoril štyri tabuľky. Tabuľky Poznamka a PoznamkaZ, ktoré slúžia pre zapisovanie si termínov, stretnutí a poznámok pre Akvizície (neživotné) a AkvizícieZ (životné). Pole IDpoznamka je primárny kľúč. Pole IDakvizice je cudzí kľúč spojený s rovnomenným poľom v tabuľke tblAkvizicie. A nakoniec tabuľky PoznamkyPS a PoznamkyPSZ, ktoré slúžia zas na zapisovanie termínov, stretnutí a poznámok pre Obchodné prípady životné a neživotné. Pole IDop je cudzí kľúč k rovnomennému poľu v tabuľke tblObchodniPrípady.

Po vytvorení všetkých tabuliek som naplnil jednotlivé tabuľky skúšobnými dátami, aby som overil, či som na nič nezabudol a hlavne, aby som odskúšal, či správne fungujú obmedzenia a vstupné masky.

3.1.2 Relácie medzi tabuľkami



Obr. 3.9 Relácie

3.2 Návrh formulárov

V ďalšom kroku som pomocou sprievodcu vytvoril z tabuliek formuláre. Následne som ich graficky upravil a pridal nejaké tlačidlá a funkcionality. Z jednotlivých formulárov som začal skladať zložené formuláre.

Číslo akvizice: {Automatické číslo} Aktualizovat

Klient: [dropdown] Pravděpodobnost získání klienta: [dropdown]
Maklér: [dropdown] Očekávaná roční provize: 0,00

Datum prvního kontaktu Datum první očekávané Datum - plná moc

Poznámky Filtr na akvizice

č. poznámky	č. akvizice	Datum	Poznámka
{Automatické}	[blue box]	[date field]	[text box]

Záznam: [navigation buttons] 1 z 1

Obr. 3.10 Formulár frmAkvizice

Formulár frmAkvizice je vytvorený z tabuľky tblAkvizice, na karte, ktorú som doňho pridal, je vložený podformulár frmPoznamky (vytvorený z tblPoznamky), ktorý je prepojený na hlavný formulár cez pole číslo akvizice, takže vidíte v ňom len poznámky k danej akvizícii. Polia Klient a Maklér sú polia so zoznamom, zdroj riadkov je dotaz (uvediem príklad: **SELECT tblKlienti.IDklient, tblKlienti.Nazev, tblKlienti.Ico FROM tblKlienti ORDER BY tblKlienti.Ico;**) a sú obmedzené na zoznam. Takže užívateľ sa pri vyplňaní nemôže pomýliť. Vedľa polí Klient a Maklér sú tlačidlá s obrázkom formuláru, po kliknutí na ne sa otvoria príslušné formuláre na vkladanie, úpravu a mazanie záznamov z príslušnej tabuľky. Tlačidlá so znakom zmazania slúžia na zmazanie jednotlivých akvizícií alebo poznámok. Nový záznam sa vkladá jednoducho, posledný záznam je vždy prázdny, stačí ho vyplniť a stlačiť tlačidlo aktualizovať alebo potvrdiť

stlačením klávesy Enter, alebo kliknutím na navigačné tlačidlá a záznam sa automaticky uloží. Úpravu záznamu môže užívateľ spraviť kedykoľvek (riešil som to tak preto, lebo zákazník si to želal čo najjednoduchšie na užívanie). Užívateľ môže vyplňať len biele polia. Polia, ktoré majú farebné pozadie, sú uzamknuté a vyplňajú sa automaticky.

Na stránke Filtr na akvizice sa nachádza vyhľadávanie pre formulár akvizície podľa makléra a podľa klienta (viď obr. 3.11). Keď budú v tomto formulári stovky záznamov, určite sa to bude hodiť. V roletke si užívateľ vyberie klienta, ktorého chce vyhľadať, stlačí tlačidlo **Vyhledat** a vo formulári mu ostanú záznamy len klienta, ktorého dal vyhľadať. Pri kliknutí na **Zobrazit vše** sa filter vypne a budú dostupné opäť všetky záznamy.

Obr. 3.11 Filtr na akvizice

Takto vyzerá procedúra pre odfiltrovanie podľa klienta:

```
Private Sub Příkaz55_Click()
    If IsNull(PoleSeSeznamem55) Then
        MsgBox "Zvolte klienta!", vbExclamation
        PoleSeSeznamem55.SetFocus
    Else
        Form_frmAkvizice.Filter = "IDklient Like '" & Me.PoleSeSeznamem55 & "'"
        Form_frmAkvizice.FilterOn = True
    End If
End Sub
```

Formulár frmAkviziceZ (teda životné) vyzerá úplne rovnako, len je prepojený s inými tabuľkami.

Veľmi podobne vyzerá aj formulár frmObchodniPripady (viď obr. 3.12) a frmObchodniPripadyZ. Rovnako ako frmAkvizice aj tieto majú podformulár s poznámkami a filtrovanie (filter je rozšírený o vyhľadávanie podľa čísla obchodnej zmluvy). Rozdiel je len v tom, že obsahuje ešte jeden podformulár navyše a to frmSplátky, ktorý je s hlavným formulárom tiež prepojený a to cez pole IDop, takže užívateľ vždy vidí splátky len ku konkrétnemu obchodnému prípadu. Inak je to veľmi podobné.

Obchodní případy

Číslo v zozname: Aktualizovat

Číslo pojistné smlouvy:

Klient:

Makléř:

Pojišťovna:

Druh pojištění:

Začátek:

Konec: Automatická prolongace: ☐

Roční pojistné: Poměrné pojistné: Provize:

Splátky **Poznámky** **Filtr na pojistné smlouvy**

Číslo splátky	Č. poj. smlouvy	Datum splatnosti	Výše splátky	Provize %	Očekávaná provize	Datum zaplacení	Zaplacená částka	Bankovní výpis
(Autom)			0,00				0,00	

Záznam: z 1

Obr. 3.12 Formulár frmObchodniPripady

Pojišťovny

Číslo pojišťovny:

Pojišťovna:

Nová pojišťovna **Upravit pojišťovnu** **Odstranit pojišťovnu**

Záznam: z 1

Obr. 3.13 Formulár frmPojistovny

Nová pojišťovna

Pojišťovna:

Storno **OK**

Upravit pojišťovnu

Číslo pojišťovny:

Pojišťovna:

Storno **OK**

Obr. 3.14 Formuláře frmPojistovnyNova a frmPojistovnyUpravit

Na obrázku obr. 3.13 je formulár frmPojistovny, ktorý je spravený pomocou sprievodcu z tabuľky tblPojistovny. Dole sú umiestnené tlačidlá, ktorými sa vykonávajú nasledovné akcie:

- Tlačidlom **Nová pojišťovna** sa otvára formulár frmPojistovnyNova (viď obr. 3.14)
- Tlačidlom **Upravit pojišťovnu** sa otvára formulár frmPojistovnyUpravit (viď obr. 3.14)
- Tlačidlom **Odstranit pojišťovnu** sa spúšťa udalostná procedúra, ktorá vymaže záznam:

```
Private Sub Prikaz6_Click()
Dim sql As String
    If MsgBox("Opravdu chcete odstranit tuto pojišťovnu?", _
vbYesNo + vbQuestion) = vbYes Then
        sql = "DELETE * FROM tblPojistovny WHERE IDpojistovna = " & _
            "Forms![frmPojistovny]![IDpojistovna]"
        DoCmd.SetWarnings (False)
        DoCmd.RunSQL (sql)
        DoCmd.SetWarnings (True)
    End If
    Form_frmObchodniPripady.IDpojistovna.Requery
    Form_frmObchodniPripadyZ.IDpojistovna.Requery
    Form_frmPojistovny.Requery
End Sub
```

Vo formulári frmPojistovnyNova sa pri stlačení tlačidla **OK** tiež spustí udalostná procedúra, ktorá pridáva záznam do tabuľky tblPojistovny:

```
Private Sub Prikaz8_Click()
Dim sql As String
    Dim na As String

    If IsNull(Nazev) Then
        MsgBox "Zadejte název nové pojišťovny!", vbExclamation
        Nazev.SetFocus
        Exit Sub
    End If

    na = "'" & Form_frmPojistovnyNova.Nazev & "'"

    sql = "INSERT INTO tblPojistovny(Nazev) " & _
        "Values(" & na & ");"

    DoCmd.SetWarnings (False)
    DoCmd.RunSQL (sql)
    DoCmd.SetWarnings (True)
    DoCmd.Close acForm, "frmPojistovnyNova"

    Form_frmObchodniPripady.IDpojistovna.Requery
    Form_frmObchodniPripadyZ.IDpojistovna.Requery
    Form_frmPojistovny.Requery

    MsgBox "Nová pojišťovna byla úspěšně přidána do databáze.", , "Nová pojišťovna"
End Sub
```

Vo formulári frmPojistovnyUpravit sa pri stlačení tlačidla **OK** zas spustí udalostná procedúra, ktorá upravuje záznam v tabuľke tblPojistovny:

```
Private Sub Prikaz8_Click()
    Dim sql As String
    Dim id As Long
    Dim na As String

    If IsNull(Nazev) Then
        MsgBox "Zadejte název pojišťovny!", vbExclamation
        Nazev.SetFocus
        Exit Sub
    End If

    id = Form_frmPojistovnyUpravit.IDpojistovna
    na = "'" & Form_frmPojistovnyUpravit.Nazev & "'"

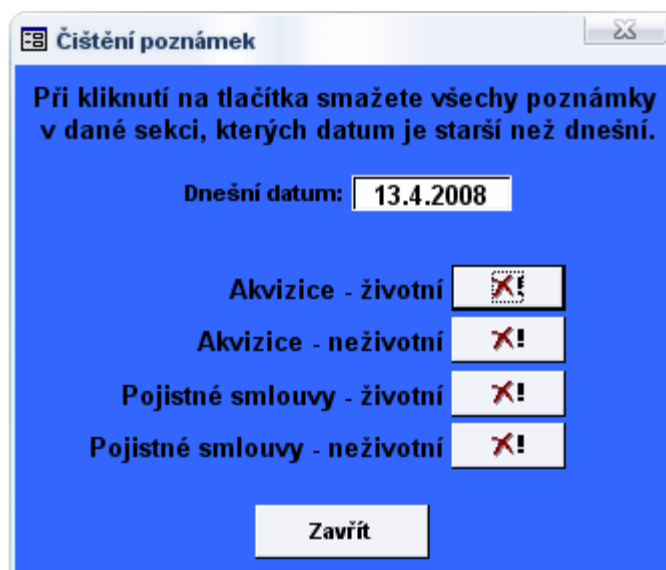
    sql = "UPDATE tblPojistovny " & _
        "SET Nazev=" & na & _
        "WHERE IDpojistovna=" & id & ";"

    DoCmd.SetWarnings (False)
    DoCmd.RunSQL (sql)
    DoCmd.SetWarnings (True)
    DoCmd.Close acForm, "frmPojistovnyUpravit"

    Form_frmObchodniPripady.IDpojistovna.Requery
    Form_frmObchodniPripadyZ.IDpojistovna.Requery
    Form_frmPojistovny.Requery

    MsgBox "Pojišťovna byla úspěšně upravena.", , "Úprava pojišťovny"
End Sub
```

Rovnakým spôsobom som riešil aj formuláre pre zobrazenie, vkladanie, úpravu a mazanie Klientov (životných aj neživotných) a Maklérov.



Obr. 3.15 Formulár CisteniePoznamok

Nakoniec som prirobil ešte jeden formulár, ktorý slúži k mazaniu starých poznámok. Keďže nepredpokladám, že užívateľ bude sám vyhľadávať a mazať staré poznámky, aby sa mu rapídne nezvyšovali nároky na pamäť, pomocou formuláru CisteniePoznamok (viď obr. 3.15) ich pár kliknutiami zmaže všetky naraz. V poli „dnešní datum“ je vložený výraz „=Date()“. Pri kliknutí na tlačidlo s obrázkom odstránenia sa spustí udalostná procedúra:

```
Private Sub PrikazO_Click()
    Dim sql As String

    If MsgBox("Opravdu si přejete odstránit všechny staré poznámky ze sekce
vbYesNo + vbQuestion) = vbYes Then
        sql = "DELETE FROM tblPoznamkaZ WHERE DatumPoznamka < Date()"
        DoCmd.SetWarnings (False)
        DoCmd.RunSQL (sql)
        DoCmd.SetWarnings (True)
    End If
    Form_frmPoznamkaZ.Requery
    Form_frmAkviziceZ.Requery
    Form_frmIndex.Requery
End Sub
```

Rovnako ako u tabuliek aj u formulárov som ich fungovanie dôkladne odskúšal.

3.3 Návrh zostáv

Zostavy som tvoril podľa potrieb zákazníka.

Prvé zostavy sú sesAkvizice (viď príloha č. 2) a sesAkviziceZ, ktoré sú vytvorené z tabuliek tblAkvizice a tblAkviziceZ. Slúžia ako prehľad akvizícií, ktoré momentálne existujú a užívateľ si ich môže v prípade potreby vytlačiť.

Ďalej som vytvoril zostavy sesPSprehled (viď príloha č. 3) a sesPSZprehled, ktoré sú vytvorené z tabuliek tblObchodniPripady a tblObchodniPripadyZ. Takisto slúžia ako prehľad existujúcich obchodných zmlúv. Pri otvorení týchto zostáv sa najprv otvorí formulár (viď príloha č. 1, obr. 8), v ktorom užívateľ vyplní dátumy, od kedy do kedy má byť koniec obchodných prípadov, ktoré chce zobraziť. Následne sa mu otvorí odfiltrovaný prehľad obchodných prípadov, ktoré vyhovujú podmienke.

Pre obchodné prípady som tiež vytvoril zostavy (pre životné aj neživotné), ktoré ich zobrazujú samostatne a so splátkami, ktoré k nim patria (viď príloha č. 4). Zostavy sú vytvorené z dotazu, ktorý obsahuje dáta z dvoch tabuliek. Pri otvorení sa tiež najprv spustí formulár (viď príloha č. 1, obr. 9), v ktorom si užívateľ vyberie klienta, od ktorého chce

vidieť zmluvy aj so splátkami, podľa neho sa potom odfiltruje zostava. Zostavy ešte navyše obsahujú celkové sumy niektorých stĺpcov splátok.

Zostavy sesSplatky (viď príloha č. 5) a sesSplatkyZ zobrazujú všetky splátky za určité obdobie (odfiltrované pomocou formuláru (viď príloha č. 1, obr. 10)). V poslednom riadku sú tiež urobené sumy niektorých stĺpcov.

Poslednou zostavou je zostava sesPoznamky (viď príloha č. 6), ktorá zobrazuje všetky poznámky za určité obdobie (odfiltrované pomocou formuláru (viď príloha č. 1, obr. 11)). Je vytvorená z dotazu, ktorý zjednocuje štyri tabuľky s poznámkami.

3.4 Dokončenie aplikácie

Najprv som vytvoril úvodný formulár (viď príloha č. 1, obr. 1), ktorý však nie je prepínací panel ani som nevybral jeden z formulárov, ale vytvoril som nový formulár, ktorý obsahuje iba kartu a stránky. Na týchto stránkach sú umiestnené jednotlivé formuláre a na poslednej stránke je niečo ako prepínací panel pre otváranie výstupov (zostáv)(viď príloha č. 1, obr. 4). Tento formulár sa otvorí vždy pri spustení aplikácie.

Následne som vytvoril vlastný panel ponúk (viď príloha č. 1, obr. 2), na ktorom sa nachádzajú iba ponuky, ktoré užívateľ bude používať (ako otváranie formulárov ...). Vytvoril som aj panel nástrojov pre zostavy (viď príloha č. 1, obr. 6), ktorý sa zobrazí len pri zostavách.

Nakoniec som v Nástrojoch nastavil, aby sa databáza pri každom vypnutí skomprimovala.

3.5 Zavedenie aplikácie u zákazníka

Zákazník má na svojom PC nainštalované balík aplikácií Microsoft Office 2007 Professional, ktorý obsahuje aj Access 2007. Ja som síce vytváral aplikáciu v Accessi 2003, no súbory z Accessu 2000, 2002, 2003 s príponou **.mdb** sa dajú v Accessi 2007 bez problémov otvoriť. Preto je zavedenie mnou vytvorenej aplikácie veľmi jednoduché, stačilo skopírovať súbor do zákazníkovho PC.

Po spustení sa však vyskytol menší problém, pretože zákazník nemal v Accessi povolené makrá. Problém sa dá odstrániť, keď v Nástrojoch Accessu v Centre zabezpečenia povolíte makrá bez upozornenia a aplikácia pôjde bez problémov a bez zbytočných upozornení.

4 Zhodnotenie prínosov navrhnutej databázovej aplikácie

Základným prínosom navrhnutej databázovej aplikácie je fakt, že ide o relačnú databázu, čo so sebou prináša hneď niekoľko zlepšení. Oproti doteraz používanému MS Excelu to znamená veľký pokrok.

Prvým prínosom je odstránenie nadbytočnosti dát. V databázovej aplikácii už totižto nedochádza k zbytočnému opakovaniu rovnakých údajov a užívateľ nemusí dookola vypisovať údaje, ktoré do databázy už raz zadal. Použitie databázovej aplikácie tiež odstraňuje problémy s násobnými hodnotami.

Veľkým prínosom je tiež ošetrovanie vzniku anomálií. Anomálie pri aktualizáciách vznikajú veľmi ľahko. Napríklad pri zmene názvu klienta by musel užívateľ ručne prepisovať všetky záznamy, v ktorých sa názov klienta nachádza, a keby sa stalo, že nejaký vynechá, došlo by ku chybe, ktorá by viedla k nesprávnym výsledkom a ďalším možným problémom. To sa však pri používaní tejto aplikácie stať nemôže. Pri vkladaní nového záznamu môže tiež dochádzať k anomáliám. Napríklad užívateľ bude vkladať novú poisťnú zmluvu a niektoré kľúčové polia nechá prázdne. To však taktiež táto aplikácia nedovolí a užívateľ je niektoré polia nútený zadať. V neposlednom rade predchádza anomáliám aj pri odstraňovaní záznamov. Keby sa užívateľ rozhodol odstrániť všetky akvizície od určitého klienta, prišiel by aj o informácie o tomto klientovi, čo sa mu však v prípade používania tejto aplikácie nepodarí.

Ďalšou výhodou je samozrejme grafické rozhranie, ktoré je intuitívne a vyzerá už úplne inak ako tabuľkový editor. V hornej časti je špeciálne vytvorený panel ponúk. Dáta majú vhodnú štruktúru a sú neporovnateľne prehľadnejšie ako v MS Exceli, či už vo formulároch alebo v zostavách, ktoré sú prispôbené priamo na tlač. Hneď po spustení sa otvorí hlavný formulár, v ktorom je takmer všetko rýchlo dostupné jednoduchým kliknutím myši.

Dostávam sa k otázke časovej náročnosti. Pre užívateľa je neporovnateľne rýchlejšie a pohodlnejšie inak otravné vkladanie dát, ktoré je však dôležitou súčasťou každej databázovej aplikácie. Takisto výstupy sú dostupné jediným kliknutím, poprípade zadáním niekoľkých parametrov. Táto aplikácia tiež obsahuje viacero spôsobov ofiltrovania záznamov, čo užívateľovi veľmi uľahčí vyhľadávanie konkrétnych záznamov.

Samozrejmosťou databázovej aplikácie je, že je viac-užívateľská. To znamená, že s ňou môže pracovať viacero užívateľov naraz. Stačí, ak umiestnia aplikáciu do zdieľanej zložky na sieti. Ešte lepším riešením je databázu rozdeliť na časť dátovú, ktorá bude uložená na serveri, a časť aplikačnú, ktorú bude mať každý užívateľ na svojom lokálnom počítači.

Aby som zhrnul prínosy jednou vetou: Táto aplikácia ušetrí užívateľovi veľa námahy a času.

Záver

Databázovú aplikáciu, ktorej tvorbu som opisoval v tejto bakalárskej práci, som vytvoril pre spoločnosť BD Finance a.s.. Hoci ide o mladú spoločnosť, kvôli zložitosti procesov a veľkosti obratu pred nedávnom pocítili potrebu vidieť informácie získané ako interne v spoločnosti, tak aj iným pohľadom a potrebu vlastniť nejaký nástroj, ktorý by transformoval dáta a znalosti do dát štruktúrovaných. Inak povedané pocítili potrebu vlastniť nejaký nástroj Business Intelligence.

Rozhodli sa pre databázovú aplikáciu, ktorá má slúžiť na evidenciu obchodných prípadov a finančné sledovanie, čo sa vlastne stalo cieľom mojej bakalárskej práce. V prvej kapitole sa dozvieme niečo o teórii relačných databáz, o základných objektoch MS Accessu, o postupe pri tvorbe databázových aplikácií v MS Accessi a čiastočne som spomenul aj programovací jazyk Visual Basic for Application. V druhej kapitole som analyzoval prostriedky, ktoré zákazník momentálne využíva pre evidenciu obchodných prípadov. Tretia kapitola zachytáva samotný postup tvorby konkrétnej databázovej aplikácie a v štvrtej kapitole stručne hodnotím prínosy tejto aplikácie pre zákazníka.

Vytýčený cieľ sa mi podarilo splniť. Vytvoril som databázovú aplikáciu, ktorá spĺňa všetky požiadavky zákazníka a doplnil som ju aj o nejaké mnou navrhnuté zlepšenia a doplnky, s ktorými zákazník súhlasil.

Aplikácia dokáže evidovať poistné zmluvy životné aj neživotné, akvizície a splátky, ktoré slúžia na následne finančné sledovanie pre top manažment spoločnosti. Zákazník jednoduchým spôsobom zistí tržby za minulý mesiac, rok, ..., ale aj výhľad tržieb do budúcnosti. Systém poznámok zas umožňuje zamestnancom evidovať si aktivity spojené so získavaním nových klientov a starostlivosť o existujúce poistné zmluvy a následne opäť jednoduchým spôsobom zistiť čo má naplánované na dnes, na budúci týždeň, atď.. A to všetko priamo v jedinej aplikácii.

Je samozrejmosťou, že časom užívaním a rastom spoločnosti príde zákazník na nové a nové požiadavky na túto databázovú aplikáciu a preto som dátovú štruktúru navrhol tak, aby už teraz uspokojila niektoré ďalšie nároky a aby bola v budúcnosti ľahko rozširiteľná. Napríklad už teraz by sa dal doplniť výstup na porovnanie efektívnosti jednotlivých maklérov.

To či spoločnosť BD Finance a.s. bude v budúcnosti využívať nástroj, ktorý som pre nich vytvoril ja nemám šancu ovplyvniť, ale snažil som ho spraviť maximálne užívateľsky prívetivý. Pri odovzdávaní aplikácie do užívania zákazníkovi mi boli síce vznesené menšie námietky na množstvo dát, ktoré musia byť zadane na vstupe, no ak chce spoločnosť pracovať s kvalitnými a plnohodnotnými výstupmi vkladaniu množstva dát sa nevyhne. Množstvo vkladanych dát závisí od výstupov, ktoré si zákazník určil sám.

Zoznam použitej literatúry:

Knihy:

1. HELD, B. *Access VBA: veľká kniha řešení*. Přel. J. Kolář. 1. vyd. Brno: Computer Press, 2006. 639 s. ISBN 80-251-1112-1.
2. MORKES, D. *Tvorba aplikací v Microsoft Access 7.0*. 1. vyd. Praha: Computer Press, 1997. 196 s. ISBN 80-85896-79-6. s. 6 – 16.
3. PERSPECTION, Inc. a ONLINE TRAINING SOLUTIONS, Inc. a CRUPI, K. a FRYE, C. *Microsoft Office XP krok za krokem*. Přel. RNDr. J. Pokorný. 1. vyd. Praha: Mobil Media a.s., 2002. 774 s. ISBN 80-86593-21-5.
4. PÍSEK, S. *Access 2003*. Praha: Grada Publishing, 2004. 124 s. ISBN 80-247-0787-X.
5. PÍSEK, S. *Access 2007: podrobný průvodce*. Praha: Grada Publishing, 2007. 197 s. ISBN 80-247-1967-2.

Elektronické zdroje:

6. *Business Intelligence* [online], [cit. 31.3.2008]. Dostupné z: http://www.shopcentrik.cz/cz/slovník/art_132/business-intelligence.aspx
7. *Řešení pro manažerské rozhodování (Business Intelligence)* [online] c2006 – 2008, [cit. 31.3.2008]. Dostupné z: <http://www.autocont.cz/sluzby-ebs-aplikace-rozhodovani.cml>
8. SLEZÁK, V. *Pojišťovací makléři: ano, či ne?* [online] 14. prosinec 2005, [cit. 31.3.2008]. Dostupné z: http://fincentrum.idnes.cz/fi_osobni.asp?r=fi_osobni&c=A051213_205618_fi_osobni_zal
9. *Teorie relačních databází: Integritní omezení* [online] 10.9. 2006, [cit. 20.2.2008]. Dostupné z: <http://www.manualy.net/article.php?articleID=15>
10. *Teorie relačních databází: Normalizace* [online] 23.11. 2007, [cit. 20.2.2008]. Dostupné z: <http://www.manualy.net/article.php?articleID=13>
11. *Teorie relačních databází: Relační model dat* [online] 12.1. 2006, [cit. 20.2.2008]. Dostupné z: <http://www.manualy.net/article.php?articleID=9>

Ostatné zdroje:

12. Interné materiály spoločnosti BD Finance a.s., *Evidence pojistných smluv.xls*

Zoznam skratiek a symbolov

- | | | |
|-----|-------|------------------------------|
| 1) | a.s. | akciová spoločnosť |
| 2) | BI | Business Intelligence |
| 3) | DBMS | Database Management System |
| 4) | DBS | Database System |
| 5) | IO | integritné obmedzenie |
| 6) | MS | Microsoft |
| 7) | RDBMS | Relation (relačný) DBMS |
| 8) | RMD | relačný model dát |
| 9) | SQL | Structured Query Language |
| 10) | SRBD | system riadenia báze dát |
| 11) | VBA | Visual Basic for Application |

Prohlášení o využití výsledků diplomové (bakalářské) práce

Prohlašuji, že

- byl(a) jsem seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo,
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně ke své vnitřní potřebě diplomovou (bakalářskou) práci užít (§ 35 odst. 3),
- souhlasím s tím, že jeden výtisk diplomové (bakalářské) práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že údaje o diplomové (bakalářské) práci, obsažené v Záznamu o závěrečné práci, umístěném v příloze mé diplomové (bakalářské) práce, budou zveřejněny v informačním systému VŠB-TUO,
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona,
- bylo sjednáno, že užít své dílo – diplomovou (bakalářskou) práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne

.....
jméno a příjmení studenta

Adresa trvalého pobytu studenta:
Jaseňová 3216/11
Žilina 010 07
Slovenská republika

Zoznam príloh:

Príloha č. 1: Obrázková príloha (obrázky obr.1 – obr. 12)

Príloha č. 2: Zostava sesAkvizice (so skúšobnými dátami)

Príloha č. 3: Zostava sesPSprehled (so skúšobnými dátami)

Príloha č. 4: Zostava sesPSZjednotlive (so skúšobnými dátami)

Príloha č. 5: Zostava sesSplatky (so skúšobnými dátami)

Príloha č. 6: Zostava sesDotazPoznaka (so skúšobnými dátami)

Príloha č. 7: CD - Evidence pojistných smluv.mdb

(chcem len upozorniť, že aplikácia na CD nie je úplne totožná s tou, ktorú som opisoval v mojej bakalárskej práci a to z dôvodu, že medzičasom došlo na žiadosť zákazníka k dolad'ovaniu aplikácie)